REMARKS

Claims 17-21, 23-26 and 28-29 are pending in the above-referenced patent application. Claims 17 and 26 are independent.

The examiner maintained the rejections of claims 17-29 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,663,012 to Shimizu et al., in view of the reference "Programming, Compiling and Executing Partially-Ordered Instruction Streams on Scalable Shared-Memory Multiprocessors" by D. K. Probst (hereinafter Probst).

The examiner also maintained the rejections of claims 17-21 and 25-26 under 35 U.S.C. §103(a) over the reference Computer Systems Design and Architecture by V. P. Heuring and H. F. Jordan (hereinafter Heuring), in view of Probst. In addition, the examiner maintained the rejections of claims 22-24 and 27-29 under 35 U.S.C. §103(a) as being unpatentable over Heuring in view of Probst and in view of Shimizu.

Specifically, the examiner stated in the August 2, 2006, Advisory Action:

> Continuation of 11. does NOT place the application in condition for allowance because: Applicants' argues in essence on pages 6-11 "...Shimizu's instructions do not describe masks to specify the particular locations of the source or destination operands..." This has not been found persuasive. The claim language, taking claim 17 as exemplary states "a local register instruction that loads one or more bytes, specified by a field of the instruction representing a mask, within a local destination register with a shifted value or another operand." This limitation, including the representation of a mask, is found in Shimizu. Shimizu illustrates in Figures 15 and 17 a local register that loads one or more bytes with a shifted value. The mask is illustrated in Shimizu's Figures 13 and 18 and described in column 7, line 36 to column 8, line 28; column 9, line 21-34; and column 11, lines 14-46. Figure 18 shows that with each type of instruction, there is an associated mask to control which data is retained or replaced. In other words, the control bits output to the selector unit in response to the instruction is a mask, since it is a pattern of bits that control "the elimination or retention of another pattern of characters, bits, or bytes (www.dictionary.com "mask" ©2000)." The claim language "...one or more bytes, specified by a field of the instruction representing a mask..." is met since the control bits are represented by the opcode in the instruction, since the instruction's opcode details which bytes in the register are being loaded and causes the control bits to be output to the select control. Applicants' arguments seem to be suggesting that the mask itself is within the instruction word, however, this is not reflected in the claim language.

Applicant amended the independent claims to make it clearer that it is the instruction itself that includes the mask, and to also clarify that the mask is a multiple-bit mask in which each bit identifies a different byte of the destination register. Support for the clarification is provided, for example, at page 11, lines 9-27 of the originally filed application. Applicant similarly amended independent claim 26. Thus, a mask, specified by a field of applicant's instruction, causes bytes of a destination register to be selectively loaded with source data. For example, as described in the originally filed application, "[the mask] 0101 loads the 1st and 3rd bytes [of the destination register] while the other bytes remain unchanged" (page 11, line 27).

As explained in applicant's previous communications, Shimizu describes a data processor and instructions that are capable of inserting and extracting data to and from optional bit area of a register, and to a control circuit therefor (col. 1, lines 11-15). Specifically, Shimizu describes two types of instructions; the GETxx instructions, and the PUTxx instructions. Every GETxx instruction, when executed on Shimizu's processor, causes a specific predetermined byte of the source operand to be placed into the lowest portion of a destination operand (FIG. 9, col. 4, line 49 to col. 5, line 14). For example, the instruction GETB0 extracts the first byte from the head of the source operand and places it in the lowest byte of the destination operand (FIG. 9A, FIG. 14, and col. 4, lines 53-57). In a similar vein, execution of one of Shimizu's various PUTxx instructions causes a portion of a source operand to be placed into a specific predetermined byte position of a destination operand (FIG. 10, col. 5, lines 15-40). For example, execution of the instruction PUTB0 causes the lowest byte of the source operand to be placed into the first byte (most significant byte) of the destination operand (FIG. 10A, FIG. 17, col. 5, lines 21-24, and col. 10, line 50 to col. 11, line 5).

However, Shimizu's instructions do not use masks to specify the particular locations of the source or destination operands. Indeed, none of the fields of any of the GETxx and/or PUTxx instructions, as shown in FIG. 7 and discussed in col. 4, lines 11-48 of Shimizu, includes a field representing a mask.

Moreover, as noted above, each of Shimizu's instructions is only capable of causing data transfer from specific bytes of a source operand to specific bytes of a destination operand. Thus,

Shimizu's instructions would have no need to use masks to identify which bytes of the operands are involved in the data transfer operation because the specific bytes of the source and destination operands affected by operation of Shimizu's instructions are implicitly identified by the instructions themselves. If it is desired to load data from different locations of the source, or into different bytes of the destination operand, it is necessary to execute different GETxx or PUTxx instructions to accomplish that.

Accordingly, Shimizu's neither discloses nor suggests at least the features of "a local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand, the field representing a mask in which each bit of the mask identifies a different byte of the destination register," as required by applicant's independent claim 17.

With respect to the other references relied upon by the examiner to reject claim 17, the examiner admitted in the May 17, 2006, Final Office Action that:

> 31. Regarding to claims 22-24 and 27-29, Heuring in view of Probst have not taught
>
> a. Wherein the local register instruction comprises a field representing a mask that specifies which byte or bytes of the destination register are affected (Applicant's claim 22);
>
> ... (Office Action, Pages 10-11, Paragraph 30)

(Applicant notes that claim 22 was cancelled in the Amendment in Reply to Action of November 28, 2005, and it is independent claim 17 that now recites a feature similar to the above-noted feature.)

Indeed, Heuring describes a RISC computer architecture and corresponding instruction set. For example, Heuring describes at pages 150 and 159-161, a shift instruction *shr*. As shown in the SRC Instruction Set summary (no page number available), none of the listed instructions includes mask that is used to identify different bytes of a register. Furthermore, as described in the Instruction Set Summary, the instruction *shr ra, rb, c3*, causes the processor to "shift rb right into ra by constant shift count c3 ≤ 31". As more particularly explained in Heuring's Table 4.5

and Table 4.10, on pages 150 and 160, respectively, the last step T7 in the execution of *shr* instruction causes the value of the operand C, which holds the shifted value held in *rb*, to be loaded into the destination register *ra* in its entirety. Thus, none of the various embodiments of the instruction *shr* as described by Heuring causes the value of an operand to be loaded into one or more bytes of a destination register as specified by a mask, as required by applicant's independent claim 17. Further, none of the RISC processor's other instructions, as listed in the SRC Instruction Set, causes a value of an operand to be loaded into one or more specified bytes of a destination register as specified by a mask. Thus, Heuring does not disclose or suggest at least the features of "a local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand, the field representing a mask in which each bit of the mask identifies a different byte of the destination register," as required by applicant's independent claim 17.

Probst discusses the performance of a large-scale shared-memory multi-processor architecture. Probst, however, does not discuss specific instructions forming the multi-processors' instruction set, and thus does not describe instructions that load one or more specified bytes within a local register with a shifted value of another operand using a multiple-bit mask included in the instructions. Accordingly, Probst does not disclose or suggest at least the features of "a local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand, the field representing a mask in which each bit of the mask identifies a different byte of the destination register," as required by applicant's independent claim 17.

Because none of Shimizu, Heuring, and Probst discloses or suggests, alone or in combination, at least the features of "a local register instruction that loads one or more bytes, specified by a multiple-bit field of the instruction, within a local destination register with a shifted value of another operand, the field representing a mask in which each bit of the mask identifies a different byte of the destination register," applicant's independent claim 17 is therefore patentable over the cited art.

Claims 18-21, 23 and 24 depend from independent claim 17 and are therefore patentable for at least the same reasons as claim 17.

Independent claim 26 describes an apparatus featuring "a command that causes the ALU to load one or more bytes, specified by a multiple-bit field of the command, within a destination register of a selected microengine with a shifted value of another one or more bytes of a source register, the field representing a mask in which each bit of the mask identifies a different byte of the destination register." For similar reasons as those provided with respect to independent claim 17, at least these features are not disclosed by the cited art. Independent claim 26 is therefore patentable over the cited art. Claim 28-29 depend from claim 26, are therefore patentable for at least the same reasons as independent claim 26.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please apply any other charges or credits to deposit account 06-1050, referencing attorney docket 10559-320001.

Respectfully submitted,

Date: Aug. 17, 2006

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21402996.doc